RECEIVED
CENTRAL FAX CENTER
SEP 07 2006

REMARKS

I. Introduction

Claims 1-9 and 11-18 remain pending in this application. No new matter has been added. In view of the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

II. The Claim Rejections Under 35 U.S.C. § 102(e) Should Be Withdrawn

Claims 1-9 and 11-18 stand rejected under 35 U.S.C. § 102(e) as unpatentable over U.S. Patent Application Publication No. 2002/0178434 to Fox et al. ("Fox"). *6/9/06 Office Action*, p. 2.

The present invention is directed to a method for utilization of a command structure representation of data file/codes. Claim 1 recites a method executed on a computing device to perform an operation on extracted elements of a software code, wherein the software code includes a command node list, a parameter list, and a handler list, comprising the steps of "generating a list of desired elements of the software code, *wherein the desired element is one of a command node element, a parameter element, and a handler function element*" and "extracting an elements from the software code" and "determining whether the extracted element is on the list of desired elements" and "*performing an operation on the extracted element when the extracted element is determined to be on the list of desired elements, wherein the operation is one of generating a command data structure representation using the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter element.*"

In contrast, Fox describes a method for automatically generating source code in a process control environment. *Fox*, p. 4, ¶ 36. Construction begins when a user invokes interactive help application 200, which automatically creates a template input spreadsheet for the project that is being created. *Id.* The user then requests source code generator 210 to create a new base source code by utilizing template source code 206 to create or modify intermediate source code 208. *Id.* Intermediate source code 208 contains special markers that indicate where to add the generated source code 212 for each type of code added. *Id.* In step 304, the user enters program specific information regarding external data source 203 to update control software 108 in process control system 100. *Id.* at p. 4, ¶ 37. The user creates lists of items to access, defining the items by the name of the list in which items are defined, the name of the program, the variable in the source code that will receive the external data, the type of data, the external reference that identifies the name of the external data item, and whether read/write code should be generated for this item. *Id.*

Step 306 includes an error checking function, which may be either customized according to functionality of the code or according to a default function. *Id.* at p. 4, ¶ 38. In step 308, the user then instructs the source code generator 210 to add source code to intermediate source code 208 using information from database 204 to define external data to be assessed. *Id.* at p. 4, ¶ 39. External data may be received from monitors and sensors throughout the process control system. In step 310, interactive help application 200, utilizing information from user input actuator 202 identifies Read or Write locations for source code generator 210 to add code that will perform a Read or Write to the process control system. *Id.* at p. 4, ¶ 40. What items to read or write are identified by the list selected by the user from database 204. *Id.* In addition, the

user can subsequently modify the information in database 204 and regenerate the source code based on the new definition, in step 312. *Id.* at p. 5, ¶ 42. Source code generator 210 will then find the appropriate code in the program source code, delete the old source code, and insert the new source code. *Id.* In step 314, Reads or Writes may be subsequently added to or deleted from the generated source code 212. *Id.* at p. 5, ¶ 43. In the final step, 316, the custom generated source code is complete. *Id.* at p. 5, ¶ 44.

In the method described by Fox, the items to read or write are selected by the user and contain specific information that identifies the external data to be assessed, such as a program variable, data type, and external reference. *See Id.* at p. 12. Thus, Fox does not show or suggest a software code that includes a "command node list, a parameter list and a handler list" or "generating a list of desired elements of the software code, *wherein the desired element is one of a command node element, a parameter element, and a handler function element*," as recited in claim 1. In addition, although the Read or Writes may be extracted from the software code, Fox does not show or suggest that they are extracted for the purpose of "*performing an operation on the extracted element when the extracted element is determined to be on the list of desired elements,*," as recited in claim 1. Furthermore, the method of Fox is specifically designed for automatic source code generation, rather than performing an operation, "*wherein the operation is one of generating a command data structure representation using the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter element*" as recited in claim 1.

Therefore, for at least the reasons stated above, it is respectfully submitted that claim 1 is not anticipated by Fox and that this rejection should be withdrawn. Because claims 2-

9 depend from and therefore include all of the limitations of claim 1, it is respectfully submitted that these claims are also allowable.

Similarly, claim 11 recites a system comprising "a first engine receiving a list of elements of a first software code, *wherein the software code includes a command node list, a parameter list, and a handler list*" along with "a second engine extracting an element from the software code" in addition to "a third engine determining whether the extracted element is on the list of desired elements" and "a fourth engine *performing an operation on the extracted element when the extracted element is determined to be on the list of desired elements, wherein the operation is one of generating a command data structure representation using the command node element, generating handler function definitions and parameter definitions using the handler function element and the parameter function element, or generating the handler function code using the handler function element and the parameter element.*"

For at least the same reasons as stated above in regard to claim 1, it is respectfully submitted that claim 11 is not anticipated by Fox and that this rejection should be withdrawn. Because claims 12-18 depend from and therefore include all of the rejections of claim 11, it is respectfully submitted that these claims are also allowable.

CONCLUSION

In light of the foregoing, Applicants respectfully submit that all of the now pending claims are in condition for allowance. All issues raised by the Examiner having been addressed. An early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

Dated: September 7, 2006

By: _____
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038